

Attorney's Docket No. FR920010023US1/9407-24

*HF*  
*IFW*  
PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re: M. Barturen et al.

Serial No.: 09/943,563

Filed: August 30, 2001

For: INTEGRATED SYSTEM AND METHOD FOR THE MANAGEMENT OF A  
COMPLETE END-TO-END SOFTWARE DELIVERY PROCESS

Examiner: Chrystine Pham

Group Art Unit: 2192

Confirmation No. 2529

Date: October 21, 2005

Mail Stop Appeal-Brief Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**APPELLANTS' BRIEF ON APPEAL UNDER 37 C.F.R. § 41.37**

This *Appeal Brief* is filed pursuant to the *Notice of Appeal to the Board of Patent Appeals and Interferences* filed September 16, 2005.

It is not believed that an extension of time and/or additional fee(s) are required, beyond those that may otherwise be provided for in documents accompanying this paper. In the event, however, that an extension of time is necessary to allow consideration of this paper, such an extension is hereby petitioned under 37 C.F.R. § 1.136(a). Any additional fees believed to be due may be charged to Deposit Account No. 09-0457.

**Real Party In Interest**

The real party in interest is assignee International Business Machines Corporation of Armonk, New York.

**Related Appeals and Interferences**

Appellants are aware of no appeals or interferences that would be affected by the present appeal.

**Status of Claims**

Claims 1-17 remain pending, each of which is finally rejected. Appellants appeal the final rejection of Claims 1-17. The attached Appendix A presents the pending claims as finally rejected in the Office Action of June 16, 2005.

### **Status of Amendments**

The attached Appendix A presents the claims as they currently stand. An *Amendment* was filed in this case on March 8, 2005 in which Claim 6 was amended and new Claims 12-17 were added. This *Amendment* was entered. A *Request for Reconsideration* (that did not include any claim amendments) was filed on August 11, 2005 in response to the Final Office Action of June 16, 2005. No *Advisory Action* was received in response to the *Request for Reconsideration*. Appellants' representative left messages with the patent examiner, but ultimately filed a *Notice of Appeal* when no *Advisory Action* was issued by the three month deadline for responding to the *Final Office Action*. Shortly thereafter, a new patent examiner contacted Appellants' representative and indicated that an *Advisory Action* would be issued in this case. However, as of the date of this *Appeal Brief*, no *Advisory Action* had been received.

### **Summary of Claimed Subject Matter**

#### **A. Independent Claim 1**

Independent Claim 1 is directed to an integrated data processing system that manages the delivery of software products to target computers or other target processing units. The language of Claim 1 is repeated below with reference numerals included that indicate the components shown in the embodiment of Figs. 1 and 2 of the present application that correspond to the components recited in Claim 1:

1. An integrated data processing system (201) for managing a process of delivery of software products to target software product execution units (103) in a network environment, comprising:

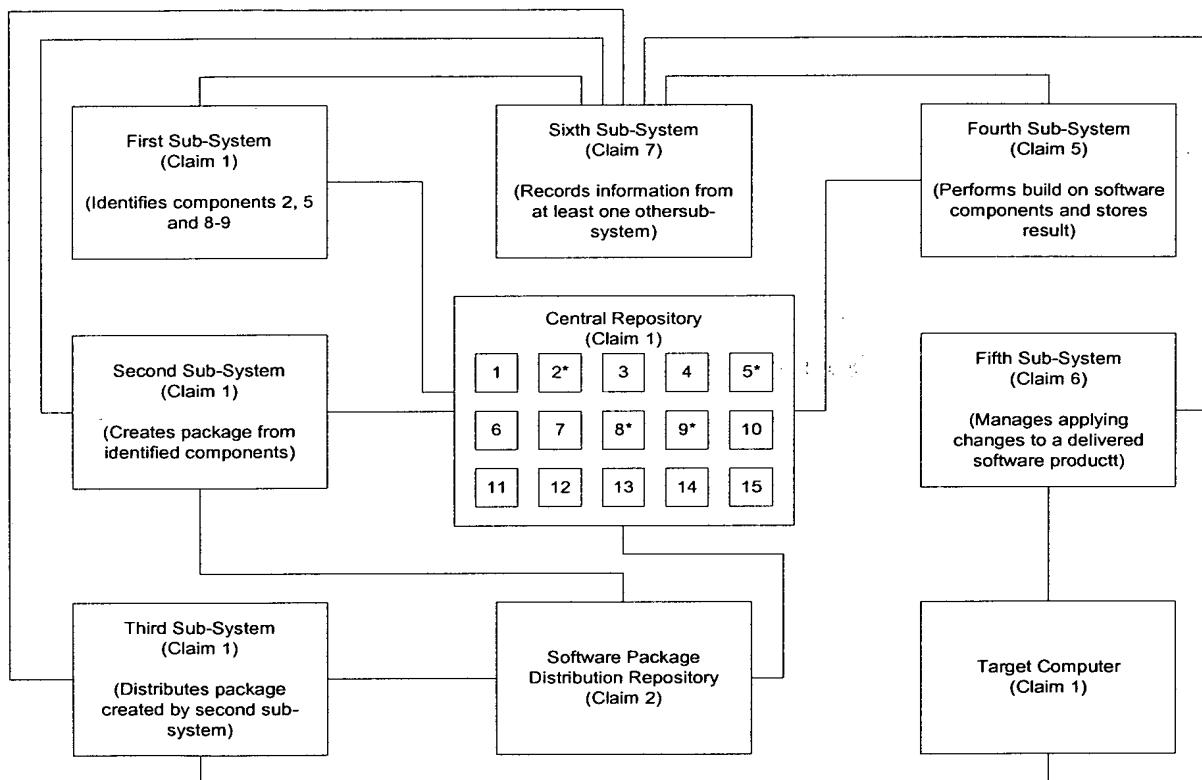
a central repository (215) for storing software components of at least one software product;

a first sub-system (203) for identifying within the central repository software components of a software product to be delivered;

a second sub-system (205) for creating at least one software product package from the identified software components identified by the first sub-system, and

a third sub-system (209) for distributing the at least one software product package created by the second sub-system (205) to the target software product execution units (103).

To further illustrate the invention of Claim 1 (and the inventions of dependent Claims 2-7 which depend from Claim 1), Appellants provide below a block diagram (which is a combined and expanded version of FIGS. 1 and 2 of the present application) of an integrated data processing system according to various embodiments of the invention of Claims 1-7. In the block diagram, all of the subsystems recited in any of Claims 1-7 are included, and connectors are provided that show how the various subsystems can be interrelated in one specific embodiment of the present invention. It will be understood that other embodiments of the invention may have fewer elements and different connections.



As shown in the block diagram above, the integrated data processing system according to Claim 1 includes a Central Repository that stores the software components required for at least one software product. These components are illustrated in the block diagram by the small numbered boxes included within the Central Repository. The integrated data processing system according to Claim 1 further includes a First Sub-System that is used to identify the software components within the Central Repository that are needed to build and/or deliver a target software product to a target end-user computer (or other execution

unit). In the exemplary block diagram, the First Sub-System has identified software components 2, 5, 8 and 9 (as indicated by the "\*" included after each of these components in the Central Repository). The integrated data processing system according to Claim 1 further includes a Second Sub-System that creates one or more software product packages using the software components identified by the First Sub-System. In certain embodiments of the present invention (see Claim 2), the software product package that is created by the Second Sub-System may be stored in a separate Software Package Distribution Repository (in other embodiments it may be stored in other locations such as, for example, the Central Repository). Finally, the integrated data processing system according to Claim 1 includes a Third Sub-System that distributes the software product package created by the Second Sub-System to the target end-user computer.

As is also shown in the block diagram above and discussed in Claim 5, in certain embodiments of the present invention, a Fourth Sub-System may be provided that performs a building process using at least some of the identified software components. The resulting components generated by the build process may then be stored, for example, in the Central Repository. In addition, in certain embodiments (see Claim 6), a Fifth Sub-System may be provided that manages the process of applying changes to software products that have already been delivered. In still other embodiments (see Claim 7), a Sixth Sub-System may be provided that records information provided by one or more of the other sub-systems.

**B. Independent Claim 8**

Independent Claim 8 is directed to a method for delivering software products to target software product execution units (boxes 103 in Fig. 1). Pursuant to the methods of Claim 8, software components of one (but often more than one) software product(s) are stored in a central repository (box 215 in Fig. 2). The software components in the central repository (215) that correspond to a specific software product that is to be delivered are then identified. Next, a software product package is created that includes at least one of the identified software components. This software product package is then distributed to, and installed on, the target software product execution units (103).

**C. Independent Claim 12**

Independent Claim 12 is directed to methods of developing and installing a software product on a group of target computers (boxes 103 in Fig. 1). Pursuant to the methods of Claim 12, a plurality of components are stored in a central repository (box 215 of Fig. 2). At least some of the stored components are used to build the software product. Once built, the software product is returned to the central repository (215). An installable software package is then created, where the installable software package includes at least some of the plurality of components and the built software product. This installable software package is stored in a second repository. The installable software package is then distributed to, and installed on, at least some of the target computers (103).

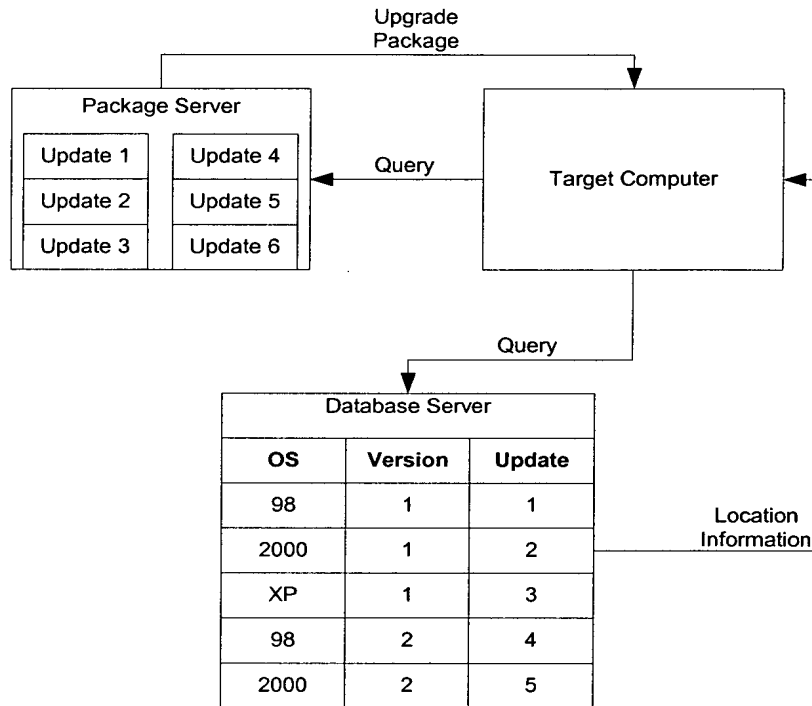
### **Grounds of Rejection to be Reviewed on Appeal**

1. The rejections of Claims 1-15 and 17 under 35 U.S.C. § 102(b) as anticipated by U.S. Patent No. 5,974,454 to Apfel et al. ("Apfel").
2. The rejection of Claim 16 under 35 U.S.C. § 103(a) as being unpatentable over Apfel in view of U.S. Patent No. 6,110,228 to Albright et al. ("Albright").

### **Argument**

#### **I. The System of Apfel**

Before addressing the pending rejections, it is helpful to compare and contrast the system disclosed in Apfel to the systems and methods of the present invention which are briefly described in the *Summary of the Claimed Subject Matter* section above. Apfel is directed to a method and system for updating software programs that are already resident on target computers. In the system of Apfel, a date on which an upgrade to a software program is expected to be available is stored in a registry on the target computer during the initial installation of the software program. About when that date arrives, the target computer may automatically contact a server over the internet to determine if an upgrade is in fact available. If it is, a process may then be initiated to install the upgrade on the target computer. Appellants have provided below an expanded version of the block diagram of FIG. 3 of Apfel which shows more detail regarding the operation of Apfel based on the detailed description of Apfel. (See Apfel at FIG. 3, Col. 2, lines 38-49, Col. 6, line 38 through Col. 7, line 9, Col. 9, lines 35-41)



As shown in the block diagram above, the system of Apfel uses a two server system to deliver a software update to a target computer. The Target Computer initiates the process by sending a query to a Database Server. This query provides the Database Server information regarding the Target Computer and the software/operating systems thereon along with a request for any available updates to a specific software package resident on the Target Computer. The Database Server conducts a search of a database resident on the Database Server to identify whether an upgrade is available and, if so, the location of the upgrade on a Package Server that stores and distributes software upgrades. The Database Server then sends this information back to the Target Computer. The Target Computer uses this information to send a second query to the Package Server requesting the specific update package identified as the appropriate package by the Database Server. The Package Server responds to this query by sending a self-installing upgrade package to the Target Computer that executes on the Target Computer to install the software upgrade.

## **II. The Rejections of Claims 1-15 and 17 Under 35 U.S.C. § 102(b) Should be Reversed**

Claims 1-15 and 17 stand finally rejected as anticipated under 35 U.S.C. § 102(b) by Apfel. (Final Office Action at 5). For the reasons discussed below, Appellants respectfully submit that the rejections of Claims 1-15 and 17 should be reversed.

### **A. The Rejection of Claim 1**

Claim 1 recites:

1. An integrated data processing system for managing a process of delivery of software products to target software product execution units in a network environment, comprising:

a central repository for storing software components of at least one software product;

a first sub-system for identifying within the central repository software components of a software product to be delivered;

a second sub-system for creating at least one software product package from the identified software components identified by the first sub-system, and

a third sub-system for distributing the at least one software product package created by the second sub-system to the target software product execution units.

The Final Office Action points to numerous different components of Apfel as allegedly disclosing each of the above recitations of Claim 1. The basis for contending that these disparate components allegedly disclose the invention of Claim 1 is far from clear, and the lack of any response to Appellants' Request for Reconsideration exacerbates the problems caused by the unclear nature of the final rejection of Claim 1.

By way of example, the Final Office Action states that the first sub-system recitation of Claim 1 is taught by Apfel's disclosure of (a) the *query 100*, (b) the *upgrade package message*, (c) *URL*, (d) the *package server*, (e) the *database query* (f) the *database lookup* and (g) the *upgrade*. (See Final Office Action at 6, listing the above components and citing to FIG. 3 and Col. 2, lines 36-50, Col. 6, lines 45-59 and Col. 7, lines 13-20 of Apfel). The Final Office Action, however, fails to explain what combination of the seven (7) separate items enumerated above allegedly constitutes the first sub-system of Claim 1 or how that unidentified combination acts as such a first sub-system.

The Final Office Action similarly cites to a list of components from Apfel as disclosing the second sub-system recitation of Claim 1, including (a) the *database query*, (b) the *versions* of the software, (c) the *platform*, and (d) the *database server*. (See Final Office Action at 6, citing to Col. 2, lines 29-47, Col. 8, lines 53-67 and Col. 9, lines 35-41 of Apfel). The Final Office Action appears to take the position that the Package Server 80b of Apfel discloses the central repository of Claim 1, and that the delivery of an upgrade from the Package Server to the target computer and the installation of the upgrade package on the computer constitutes the third sub-system of Claim 1. (Final Office Action at 6).

For at least the reasons discussed in the following sections, Appellants respectfully submit that Apfel does not disclose or suggest the system of Claim 1.

1. Apfel Does Not Disclose the Second Sub-System of Claim 1

As an initial matter, Apfel fails to disclose or suggest a "second sub-system for **creating at least one software product package from the identified software components** identified by the first sub-system." The portions of Apfel cited in the Final Office Action simply indicate that a variety of different upgrade packages may be stored on the Package Server to account for the fact that different operating systems, different languages and different versions of the software application may require different upgrade packages. (See Apfel at Col. 9, lines 35-41; *see also* Col. 2, lines 29-47 and Col. 8, lines 53-67). The Database Server of Apfel keeps track of the plurality of different upgrade packages that are provided for each software application so that it can notify the target computer as to the location of the appropriate upgrade on the Package Server. (*Id.*). However, there is no teaching in Apfel that the Database Server "create[s] . . . [the] software product package" as does the Second Sub-System of Claim 1, let alone create such a software product package "from the . . . software components identified by the first sub-system" as is recited in Claim 1. Instead the Database Server merely performs a database lookup operation. As such, Apfel fails to disclose or suggest the claimed Second Sub-System, and hence the rejection of Claim 1 should be reversed for at least this reason.

Appellants note that the Response to Arguments section of the Final Office Action provides further insight into the patent examiner's rationale for claiming that Apfel discloses the second sub-system of Claim 1. Specifically, the Final Office Action states:



In col. 9:35-41, Apfel discloses using information received in the database query (i.e., versions of the program components, and operating system on which the components are running) to determine whether upgrade package is available. In the same passage, Apfel further discloses providing different update packages for different version combinations, different operating systems. ***It is clear that the update package for different version combinations and different operating systems*** (as identified by the database query or "first sub-system") ***have to be created in order to meet the update demand*** of target software execution units.

(Final Office Action at 3, underline in original, bold and italics added). Appellants respectfully submit that the concluding sentence in the argument quoted above does not follow either from the sentences that precede it, nor is it supported by the disclosure of Apfel. More importantly, what Claim 1 recites is that the software product package is created "from the . . . software components identified by the first sub-system." While it appears that the Final Office Action may take the position that the query 100 of Apfel identifies "software components" that are used to create the "at least one software product package", this is not the case.<sup>1</sup> Apfel never discloses or suggests that the different update packages are created from software components that are identified by the database query. Instead, Apfel teaches that the query is made to determine if an update of a software package that runs on a specific operating system is already available. (See, e.g., Apfel at Abstract, noting that a "database lookup" is performed in response to the query to see if an updated software program is currently available). Thus, as Apfel does not disclose or suggest creating a software product package from software components that are identified by a first sub-system, the rejection of Claim 1 should be reversed.

## 2. Apfel Does Not Disclose the First Sub-System of Claim 1

Appellants also respectfully submit that the rejection of Claim 1 should be reversed because Apfel further does not disclose or suggest "a first sub-system for identifying within the central repository software components of a software product to be delivered" as recited in Claim 1. The Final Office Action states that the database query 100 of Apfel comprises the first sub-system. (See, e.g., Final Office Action at 2). Even were this correct, the query

---

<sup>1</sup> Specifically, the Final Office Action states that it "is clear that the update package for different version combinations and different operating systems (as identified by the database query or "first sub-system") have to be created in order to meet the update demand of target software execution units." (Final Office Action at 3).

100 does not "identify . . . software components of a software product " as recited in Claim 1 – instead it provides information which the Database Server uses to identify an appropriate upgrade package (i.e., a software product as opposed to the components thereof). Accordingly, Apfel's failure to disclose or suggest the first sub-system recitation of Claim 1 provides a further basis for reversal of the pending rejection of Claim 1.

## **B. The Rejection of Claim 2**

Dependent Claim 2 is patentable over Apfel for at least the reasons that Claim 1 is patentable. In addition, Claim 2 adds the recitation that the integrated data processing system further includes a "software package distribution repository" that stores at least one of the software product packages that were created from the identified software components. The Final Office Action cites to the Package Server of Apfel as disclosing this recitation. (Final Office Action at 6-7). However, the Final Office Action also cites to the Package Server of Apfel as comprising the Central Repository. Appellants respectfully submit that the examiner cannot have it both ways – i.e., if the product updates stored in the Package Server of Apfel comprise "software components of at least one software package" as argued in the rejection of Claim 1, those same product updates cannot also comprise "software product packages" that were created from the software components. As such, Claim 2 is independently patentable over Apfel for at least these additional reasons.

Appellants note that in the Final Office Action the Examiner argues that the Package Server stores both (1) "upgrade packages" which are identified as "at least upgrade, program modules, query, versions of program module components, Web Authoring Components Program Module, word processor program module and HTML converter" and (2) "software components" which are identified as "program module components" (citing to Col. 6, lines 18-25 of Apfel), which are argued to be something different. (See Final Office Action at 4). This, however, is simply not the case. Instead, Apfel makes clear that the "program modules", "program module components", "versions of program module components", the "Web Authoring Components program module", the "word processor module" and the "HTML converter" of Apfel are all software programs that may be running on a target computer. These software programs are generally referred to as "program module components" in Apfel (or sometimes by the shortened notation, "program modules"), and it is upgraded versions of these program modules that are stored on the Package Server.

References to "versions of program module components" reflects that a particular software program may have multiple versions. The "Web Authoring Components program module", the "word processor module" and the "HTML converter" are all specific program module components (i.e., software programs) that are provided as examples in Apfel. As the above discussion makes clear, the rejection of Claim 2 is pointing to the same thing – the upgrades stored on the Package Server – as comprising both the "software components" and the "software product packages" of Claim 2. As such, the arguments raised in the Final Office Action do not rebut Appellants showing that the rejection of Claim 2 is improper.

### **C. The Rejection of Claim 3**

Claim 3 recites that the software product packages are distributed to execution units that belong "to at least one environment according to at least one role assigned to the at least one software product package by the second sub-system." As noted above, Apfel does not disclose or suggest the claimed second sub-system, let alone a sub-system that assigns roles to the software product packages. Accordingly, Claim 3 is independently patentable over Apfel for at least this additional reason.

In response to the rejection of Claim 3, the Examiner argues that "each software product package is associated with an operating system on which it is to run (i.e., 'assigned a role')." (Final Office Action at 4). Assuming, for purposes of argument, that this statement is correct, it still does not establish that Apfel discloses or suggests that the "at least one role" is "assigned to the at least one software product package by the second sub-system" as recited in Claim 3. Appellants submit that no such teaching is provided in Apfel, because it is axiomatic that the Package Server of Apfel does not assign environments or operating systems to the target computers.

### **D. The Rejection of Claim 5**

Claim 5 recites a fourth sub-system that performs a building process on selected ones of the software code components. The Final Office Action cites to Col. 9, lines 35-41, step 451 of Fig. 4B and Col. 10, lines 61-63 of Apfel as disclosing the recitations added by Claim 5. (Final Office Action at 7). Appellants respectfully disagree with this finding. In particular, as noted above, the passage of Apfel at Col. 9, lines 35-41 simply states that a variety of different upgrade packages are maintained on the Package Server and that the

Database Server keeps track of the different available upgrades using a database of upgrade packages. This passage of Apfel provides no teaching whatsoever that a fourth sub-system is provided or that a building process is performed on selected software code components. Likewise, step 451 of Fig. 4B and Col. 10, lines 61-63 of Apfel simply state that the upgrade package is downloaded to the computer. Once again, this has nothing to do with the recitations of Claim 5. Accordingly, Claim 5 also is independently patentable over Apfel for at least these additional reasons.

#### **E. The Rejection of Claim 6**

Claim 6 recites that the integrated data processing system includes a "fifth sub-system for managing a process of applying changes to an already delivered software product." The Final Office Action cites to passages from Apfel that discuss how the system of Apfel may be used to provide an upgrade package to a target computer. However, the fifth sub-system recited in Claim 6 manages a process of applying changes to the software product package that is distributed by the third sub-system recited in Claim 1. Apfel does not disclose or suggest such a sub-system. Instead the upgrade package is delivered to the target computer and it either will or will not work – no sub-system is provided for managing the process of applying changes to the delivered product package. Accordingly, Claim 6 is also independently patentable over the cited art for at least these additional reasons.

With respect to the rejection of Claim 6, the Final Office Action states that "a process of applying changes to an already delivered software product" merely refers to the process of installing an upgrade package on the target computer. (Final Office Action at 5). Appellants respectfully submit that "installing" and "applying changes to" a software product are two very different things, as is made clear in the specification of the present application. (*See, e.g.*, Specification at 12-13 and 20). Accordingly, Claim 6 is also independently patentable over the cited art for at least these additional reasons.

#### **F. The Rejection of Claim 8**

Claim 8 is directed to a method for delivering software products. The Final Office Action states that Claim 8 is rejected for the same reasons that Claim 1 is rejected. (Final Office Action at 8). However, as discussed above, Apfel does not disclose or suggest either (1) "identifying software components of a software product to be delivered among the

software components stored in the central repository" or (2) "creating at least one software product package that includes at least one of the identified software components" as recited in Claim 8. Accordingly, the rejection of Claim 8 should be reversed for at least these reasons.

**G. The Rejection of Claim 9**

Claim 9 depends from Claim 8 and hence is patentable for each of the reasons, discussed above, that Claim 8 is patentable over Apfel. In addition, as noted in the Final Office Action, Claim 9 is a method claim that generally correspond to the system of Claim 3. Accordingly, Claim 9 is also independently patentable for at least the reasons, discussed above, that Claim 3 is patentable over Apfel.

**H. The Rejection of Claim 10**

Claim 10 depends from Claim 8 and hence is patentable for each of the reasons, discussed above, that Claim 8 is patentable over Apfel. In addition, as noted in the Final Office Action, Claim 10 is a method claim that generally correspond to the system of Claim 2. Accordingly, Claim 10 is also independently patentable for at least the reasons, discussed above, that Claim 2 is patentable over Apfel.

**I. The Rejection of Claim 11**

Claim 11 depends from Claim 8 and hence is patentable for each of the reasons, discussed above, that Claim 8 is patentable over Apfel. In addition, as noted in the Final Office Action, Claim 11 is a method claim that generally correspond to the system of Claim 5. Accordingly, Claim 11 is also independently patentable for at least the reasons, discussed above, that Claim 5 is patentable over Apfel.

**J. The Rejections of Claim 12-13 and 17**

Claim 12 stands rejected in the Final Office Action as anticipated by Apfel. The rejection merely incorporates the rejection of Claims 1 and 2. Accordingly, the rejection of Claim 12 should be withdrawn for each of the reasons, discussed above, that the rejections of Claims 1 and 2 should be withdrawn. Appellants further note that Apfel clearly does not teach (1) storing a plurality of components in a central repository, and then (2) building a "built software product" using the software components and storing the "built software product" back in the central repository and then (3) creating an installable software package

that includes the built software product and storing this installable software package in a second repository.

The rejections of Claims 13 and 17 should be reversed for the same reasons that the rejection of Claim 12 should be reversed, as Claims 13 and 17 each depend from Claim 12.

#### **K. The Rejection of Claim 14**

Claim 14 depends from Claim 12 and hence is patentable for each of the reasons, discussed above, that Claim 12 is patentable over Apfel. In addition, Claim 14 stands rejected based on the same rationale as Claim 6. (Final Office Action at 9). Accordingly, Claim 14 is also independently patentable for at least the reasons, discussed above, that Claim 6 is patentable over Apfel.

#### **L. The Rejection of Claim 15**

Claim 15 depends from Claim 12 and hence is patentable for each of the reasons, discussed above, that Claim 12 is patentable over Apfel. Claim 15 recites that the method of Claim 12 "further compris[es] recording information regarding the software product in a tracking sub-system." The Final Office Action states that this is disclosed at Col. 2, lines 50-61 of Apfel. (Final Office Action at 9-10). However, what the cited portion of Apfel in fact states is that the installed upgrade package includes a new date embedded there in (in a "registry key") that is used to determine when in the future to check for additional updates. The inclusion of a date in the upgraded software package itself clearly does not comprise "recording information regarding the software product in a tracking sub-system." Accordingly, the rejection of Claim 15 should also be reversed for at least this additional reason.

### **III. The Rejections of Claim 16 Under 35 U.S.C. § 103 Should be Reversed**

Claim 16 stands rejected under 35 U.S.C. § 103 as obvious over Apfel in view of Albright. As Claim 16 depends from Claim 12, Claim 16 is patentable as depending from a patentable base claim.

### **IV. Conclusion**

In light of the above, Appellants submit that each of the pending claims is patentable over the cited references and, therefore, request reversal of the rejections of Claims 1-17.

In're Barturen et al.  
Serial No. 09/943,563  
Filed August 30, 2001  
Page 15

Respectfully submitted,



D. Randal Ayers  
Registration No. 40,493

USPTO Customer No. 20792  
Myers Bigel Sibley & Sajovec, P.A.  
Post Office Box 37428  
Raleigh, North Carolina 27627  
Telephone: (919) 854-1400  
Facsimile: (919) 854-1401

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Mail Stop Appeal-Brief Patents, Commissioner for Patents, P.O. box 1450, Alexandria, VA 22313-1450, on October 21, 2005.



---

Carey Gregory  
Date of Signature: October 21, 2005.

## **CLAIMS APPENDIX**

**Pending Claims USSN 09/943,563  
Filed August 30, 2001**

1. (Original) An integrated data processing system for managing a process of delivery of software products to target software product execution units in a network environment, comprising:

- a central repository for storing software components of at least one software product;
- a first sub-system for identifying within the central repository software components of a software product to be delivered;
- a second sub-system for creating at least one software product package from the identified software components identified by the first sub-system, and
- a third sub-system for distributing the at least one software product package created by the second sub-system to the target software product execution units.

2. (Original) The integrated data processing system according to claim 1, further comprising a software package distribution repository for storing the at least one software product package created by the second sub-system from the identified software components.

3. (Original) The integrated data processing system according to claim 1, in which the third sub-subsystem distributes the at least one software product package to target software product execution units belonging to at least one environment according to at least one role assigned to the at least one software product package by the second sub-system.

4. (Original) The integrated data processing system according to claim 1, in which the first sub-system manages a storage in the central repository of the software components of the software product to be delivered.

5. (Original) The integrated data processing system according to claim 1, further comprising a fourth sub-system for performing a building process of software code components among the identified software components of the software product to be delivered, the fourth sub-system storing a result of the building process in the central repository.



6. (Previously Presented) The integrated data processing system according to claim 1, further comprising a fifth sub-system for managing a process of applying changes to the at least one software product distributed by the third sub-system.

7. (Original) The integrated system according to claim 1, further comprising a sixth sub-system for recording information provided by at least one of the first through fifth sub-systems of the integrated data processing system during delivery of the software product.

8. (Original) A method for delivering software products to target software product execution units in a network environment, comprising the steps of:

storing software components of at least one software product in a central repository;  
identifying software components of a software product to be delivered among the software components stored in the central repository;

creating at least one software product package that includes at least one of the identified software components;

distributing the software product package to the target software product execution units and installing the software product package thereon.

9. (Original) The method according to claim 8, in which the step of creating at least one software product package includes assigning to the at least one software product package an indication of role for identifying the target software product execution units to which the software product is to be distributed, and distributing the at least one software product package according to the indication of role.

10. (Original) The method according to claim 8, further comprising a step of storing the at least one software product package in a software distribution repository.

11. (Original) The method according to claim 10, further comprising a step of building identified source code components of the software product to be delivered stored in the central repository, and storing the result of the building in the central repository.

12. (Previously Presented) A method of developing and installing a software product on a plurality of target computers, the method comprising:

- storing a plurality of components in a central repository;
- using at least some of the plurality of stored components to build the software product;
- storing the built software product in the central repository;
- creating an installable software package that includes at least some of the plurality of components and the built software product;
- storing the installable software package in a second repository;
- distributing the installable software package to at least some of the plurality of target computers; and
- installing the distributed installable software package on the at least some of the plurality of target computers.

13. (Previously Presented) The method of Claim 12, wherein the software product comprises a newly developed software product.

14. (Previously Presented) The method of Claim 12, wherein the software product comprises a new release and/or a new version of an already released software product.

15. (Previously Presented) The method of Claim 12, further comprising recording information regarding the software product in a tracking sub-system.

16. (Previously Presented) The method of Claim 12, wherein the built software product comprises execution code that is generated from a source code component stored in the central repository.

17. (Previously Presented) The method of Claim 12, further comprising providing a configuration management subsystem that controls and manages different versions of the software components stored in the central repository.

**EVIDENCE APPENDIX**

No evidence is being submitted with this Appeal Brief pursuant to 37 C.F.R. §§  
1.130, 1.131 or 1.132.

Appeal Brief Appendices

USSN 09/943,563

Filed: August 30, 2001

Page 5

#### **RELATED PROCEEDINGS APPENDIX**

There are no related proceedings.